A Pure Logic-Based Approach to Natural Reasoning

Abzianidze, Lasha

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (APA):
Abzianidze, L. (2015). *A Pure Logic-Based Approach to Natural Reasoning*. 40-49. Paper presented at Amsterdam Colloquium 2015, Amsterdam, Netherlands.

A Pure Logic-Based Approach to Natural Reasoning

Lasha Abzianidze

TiLPS, Tilburg University, Tilburg, The Netherlands
L.Abzianidze@uvt.nl

Abstract

The paper presents a model for natural reasoning that combines theorem proving techniques with natural logic. The model is a tableau system for a higher-order logic the formulas of which resemble linguistic expressions. A textual entailment system LangPro, an implementation of the model, represents a tableau-based prover that directly operates on linguistic expressions. After training and evaluating on a textual entailment dataset, the prover shows accuracy comparable to the state-of-the-art results with almost perfect precision. Due to its reliable judgements, the system is also able to detect dubious problems in the dataset.

1 Introduction

A task of recognizing textual entailments (RTE) is one of the popular ways of testing reasoning capacity of NLP systems in natural language. RTE is usually a 3-way classification problem where a pair of T text and H hypothesis is classified based on whether T entails, contradicts or is neutral to H. The judgements shared by majority of human annotators are usually considered as a gold answer to an RTE problem. Hence the task is considered to evaluate the systems on human reasoning over natural language text.

While humans reason over natural language expressions, they heavily relay on the meaning of the expressions and use their reasoning capacity for drawing conclusions. The situation in RTE challenges is somewhat different from this. The systems are more concentrating on learning regularities from the data and later an RTE problem is classified based on how much it fits in the learned regularities. This type of approaches are usually robust but also imprecise. Their crucial drawback is inability of composing meanings of several facts and making conclusions based on them. On the other hand, approaches based on some logical language are brittle as translating from natural language into the formal language itself represents a difficult problem.

In this paper, we opt for a formal logic-based account for natural reasoning where the translation problem is facilitated with a highly expressive language. In this way, we mainly concentrate on the reasoning part and account for wide-coverage natural reasoning over linguistic text. First, the formal language and its calculus will be introduced. Then their extensions will be outlined based on which a theorem prover is constructed. We describe how the prover reasons over natural language expressions and demonstrate its performance over the SICK RTE [14] data set. The paper ends with a conclusion and the words about future work.

2 An analytic tableau system for natural logic

An analytic tableau system for natural logic [18] incorporates two main ideas according to which there are a formal semantic representation that resembles the linguistic surface form and a tableau proof system that acts on the structures of this representation. In this way, Muskens

$$\frac{\texttt{every}\ A\ B : [\,] : \mathbb{T}}{A : [d] : \mathbb{F} \qquad B : [d] : \mathbb{T}}\forall_{\mathbb{T}} \text{ where } d \text{ is } \textit{old} \qquad \frac{\texttt{who}\ A\ B : [\vec{C}] : \mathbb{F}}{A : [\vec{C}] : \mathbb{F} \qquad B : [\vec{C}] : \mathbb{F}}\wedge_{\mathbb{F}}$$

$$\frac{\texttt{some}\ A\ B : [\,] : \mathbb{T}}{\begin{array}{l}A : [c] : \mathbb{T}\\ B : [c] : \mathbb{T}\end{array}}\exists_{\mathbb{T}} \text{ where } c \text{ is } \textit{fresh} \qquad \frac{\texttt{who}\ A\ B : [\vec{C}] : \mathbb{T}}{\begin{array}{l}A : [\vec{C}] : \mathbb{T}\\ B : [\vec{C}] : \mathbb{T}\end{array}}\wedge_{\mathbb{T}} \qquad \frac{A\ B : [\vec{C}] : \mathbb{X}}{A : [B, \vec{C}] : \mathbb{X}}\text{PUSH}$$

$$\frac{\begin{array}{l}A : [\vec{C}] : \mathbb{T}\\ B : [\vec{C}] : \mathbb{F}\end{array}}{\times}\leq\!\times \text{ where } A \leq B \qquad \frac{\begin{array}{l}A : [\vec{C}] : \mathbb{T}\\ B : [\vec{C}] : \mathbb{T}\end{array}}{\times}\text{DISJ}\times \text{ where } A \mid B$$

Figure 1. The rules that are used by the tableau in Figure 2

in [18] regards natural language as a formal logical language and models natural reasoning in the same style as it is done for formal logics in terms of proof systems.[1]

The adopted semantic representation in [18] is a sort of functional type logic—the language of simply typed λ-calculus where terms are interpreted as functions.[2] For example, a term of type (α, β) is interpreted as a total function from objects of type α to objects of type β. Linguistic expressions are modelled by the terms, called Lambda Logical Forms (LLFs), built up mainly from lexical terms. Examples of LLFs are given below:

Some runner who adores Mary won (1)

$$\texttt{some}_{(et)(et)t}\ \left(\texttt{who}_{(et)(et)et}\ \left(\texttt{adore}_{eet}\ \texttt{Mary}_{e}\right)\ \texttt{runner}_{et}\right)\ \texttt{won}_{et} \tag{1a}$$

Apart from resembling the surface forms, LLFs are comparable to the logical forms of generative grammar [10], the abstract terms of abstract categorial grammar (ACG) [9], or the terms built from multi-dimensional signs of λ-grammar [17].

A tableau system for natural logic, in short a natural tableau, is a signed tableau over LLFs. A tableau entry has three components: an LLF, a list of argument terms and a sign which is either true $\mathbb{T}$ or false $\mathbb{F}$. For instance, $\texttt{love} : [\texttt{Mary}, \texttt{John}] : \mathbb{T}$ is a well-formed entry (i.e. node) of a natural tableau. The semantics behind the entry is that a term resulted from applying an LLF to the arguments in a list order, `love Mary John`, is evaluated as the truth value of the attached sign, i.e. as true in this case.

The natural tableau comes with an inventory of inference rules. The rules are used to decompose complex nodes into shorter ones. Since the tableau aims to model natural reasoning, the inventory is expected to contain a plethora of rules. Several tableau rules for Boolean operators, determiners and lexical items with certain algebraic properties (including monotonicity) are already presented in [18]. Some of those rules are given in Figure 1.[3] The intuition behind the rules is simple; for example, the rule $\forall_{\mathbb{T}}$ asserts that if every A does B, then d is

[1] At first glance this decision is in the same vein as Montague's proposal in [15], but [18] gives the calculus over a formal language in contrast to Montague's solution to translate the English sentences into a formal logic.

[2] Actually [18] employs a relational type logic since interpreting terms as relations is more intuitive and simple according to [16]. Since entailment relations of both functional and relational type logics are the same [16], here we use a more common interpretation that is the functional one. While the original work [18] considers a type system with three basic types e, s and t (corresponding to entities, possible worlds and truth values, respectively), we will omit s type as no examples involving intentionality is discussed in this paper.

[3] While presenting the rules, we assume that: $\vec{C}$ stands for a sequence of terms; $A \leq B$ denotes $\forall\vec{X}(A\vec{X} \to B\vec{X})$ which informally says that A entails or is subsumed by B; and $A \mid B$ denotes $\neg\exists\vec{X}(A\vec{X} \wedge B\vec{X})$ which means

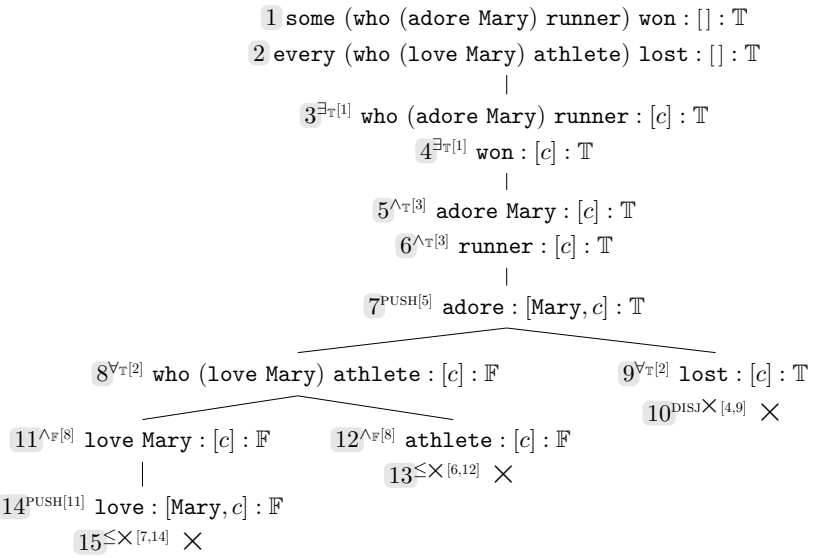

Figure 2. The tableau proving that *some runner who adores Mary won* contradicts *every athlete who loves Mary lost*. The nodes are enumerated and annotated with a source rule application.

not A or d does B given that d is some entity in a considered situation. With the help of the rules, it is possible to check LLFs on consistency and hence for entailment and contradiction too. For example, to check whether LLF_1 contradicts LLF_2, it is sufficient to check whether LLF_1 and LLF_2 together are consistent. The latter is done by starting a tableau with entries $LLF_1 : [\vec{C}] : \mathbb{T}$ and $LLF_2 : [\vec{C}] : \mathbb{T}$, where $\vec{C}$ is a sequence of terms feeding LLFs till the truth values. In this way, the tableau in Figure 2 proves inconsistency of sentences by showing that they are true together in no possible situation, i.e. all tableau branches are closed. All the rules used in the tableau can be found in Figure 1.

3 A wide-coverage natural language prover

3.1 Extending the natural tableau

For a wide-coverage tableau system we decide on what LLFs should look like for unrestricted sentences and also extend the format of tableau entries. We obtain LLFs automatically from the Combinatory Categorial Grammar (CCG) derivations [1, 2]. While doing so, it is possible to obtain LLFs typed with non-directional syntactic types prior to the final LLFs of semantic types. From the perspective of the tableau system, LLFs with syntactic types offer better matching between tableau nodes and antecedents of the rules than LLFs with semantic types.[4] Due to this reason, in the extended version of the tableau system LLFs are typed with syntactic

that A and B are disjoint concepts. The subsumption and disjoint relations for lexical constants are assumed to be a part of background knowledge.

[4]Note that an LLF of type et can ambiguously correspond to a bare singular noun phrase or an intransitive verb phrase. But further decomposition of the LLF in a tableau itself requires resolving this ambiguity which complicates the whole process. On the other hand, an LLF of syntactic type contains no such kind of ambiguity.

types. For instance, the LLF of syntactic type for (1) is (1b) which differs from (1a) only in types.[5] Note also that new LLFs now more resemble the abstract terms of ACG [9] and the signs of λ-grammar [17] due to their syntactic types.

$$\mathbf{some}_{\mathtt{n},\mathtt{vp},\mathtt{s}}\ (\mathbf{who}_{\mathtt{vp},\mathtt{n},\mathtt{n}}\ (\mathbf{adore}_{\mathtt{np},\mathtt{vp}}\ \mathbf{Mary}_{\mathtt{np}})\ \mathbf{runner}_{\mathtt{n}})\ \mathbf{won}_{\mathtt{vp}} \tag{1b}$$

Terms of semantic types are still needed while modelling certain phenomena. For example, an intersective adjective *red* is modelled as a term of type $(\mathtt{n},\mathtt{n})$, but additionally a term of semantic type et is also necessary to assert redness of an object. Introduced fresh entity terms are also of type e. In order to accommodate both terms of syntactic type and semantic type in the same language, [2] introduces a subtyping relation over syntactic and semantic types. For example, given that $e \sqsubseteq \mathtt{np}$, $\mathtt{s} \sqsubseteq t$ and $\mathtt{n} \sqsubseteq et$, terms like $\mathbf{love}_{\mathtt{np},\mathtt{vp}}\,\mathbf{Mary}_{\mathtt{np}}\,c_e$ and $\mathbf{athlete}_{\mathtt{n}}\,c_e$ are well-typed terms of the new language with the extended type system.

Another extension to the natural tableau is to add a modifier set in tableau entries. As argued in [2], the set is used to save a modifier term that is indirectly applied to its head. A modifier is discharged from the set and applied to the main LLF of an entry when the LLF is a lexical term. This technique is used for the nouns with several adnominals or verbs with several adverbs. The trick with the modifier set solves the problem of event modification without introducing an event variable or the existential closure operator in LLFs, opposed to the approach in ACG [20]. Note that the extension is conservative in the sense that the tableaux generated with [18], e.g., the one in Figure 2, are still available in the extended version. This is done by considering entries with the old format as having an empty modifier set.

3.2 A theorem prover for natural language

Our next step is to present a theorem prover for natural logic [1] implemented based on the extended natural tableau system. The natural logic prover, like its theoretical model, has a modular architecture involving 3 main components: a knowledge base (KB), an inventory of rules (IR), and a proof engine (PE). The KB contains the hyponymy/hypernymy and antonymy relations of WordNet [8] as facts. The IR consists of around 80 rules part of which are taken from [18] and the rest are designed manually based on RTE training datasets [2]. Apart from new rules for passives, modifier-head constructions, copula, expletives, prepositional phrases, etc., there are also *admissible* (i.e. shortcut) rules in the IR which contribute to shorter proofs.

In order to reason over natural language expressions in an automatized way, the prover is paired with one of the state-of-the-art CCG parsers, C&C [6] or EasyCCG [11], and the LLF generator. The latter module produces LLFs from a CCG derivation by first replacing the CCG categories with syntactic types, then correcting inadequate analyses and finally type-raising determiners. The whole pipeline results in a theorem prover for natural language, called LangPro. The chart in Figure 3 shows how LangPro works.[6]

Compared to the forefront RTE system Nutcracker [3], based on first-order logic theorem provers and model builders, LangPro employs more expressive higher-order logic. As a result our system models intersective adjectives and higher-order quantifiers properly in contrast to Nutcracker. On the other hand, LangPro is backed up by a formal logical language opposed to NatLog [12], a prominent RTE system motivated by natural logic. Due to this reason NatLog is not able to reason over several premises and cannot account for logical laws like of De Morgan.

[5] An employed set of atomic types $\{\mathtt{n},\mathtt{np},\mathtt{pp},\mathtt{s}\}$ corresponds to basic syntactic types of CCG for noun, noun phrase, prepositional phrase and sentence, respectively. Often $(\mathtt{np},\mathtt{s})$ type is abbreviated as $\mathtt{vp}$. Hereafter, a lexical term of syntactic type is denoted by its lemma and is written in boldface.

[6] See the online demo of LangPro at: `http://lanthanum.uvt.nl/labziani/tableau`

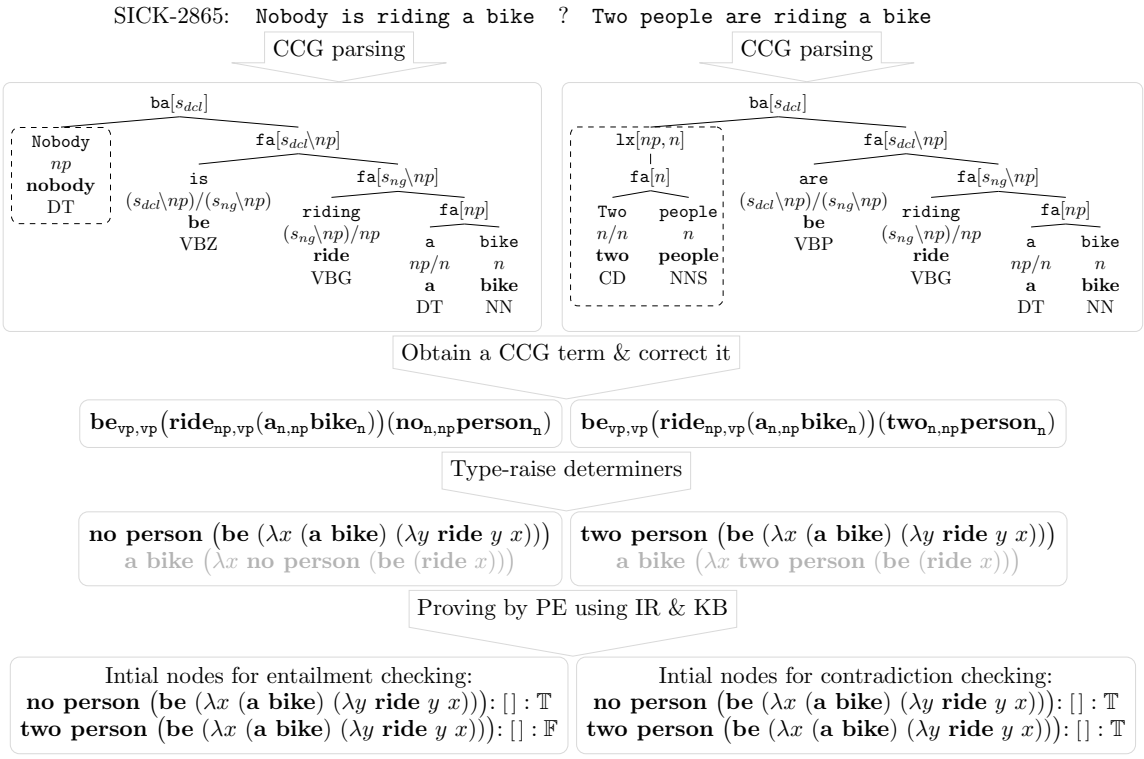

Figure 3. LangPro processing an RTE problem. To maintain the process efficient only one LLF from possibly several LLFs is employed by the tableau prover.

4 Adapting to SICK

4.1 Several new rules

The SICK (Sentences Involving Compositional Knowledge) dataset [14] is a set of about 10K text-hypothesis pairs annotated with three labels: entailment, contradiction and neutral.[7] The dataset is divided in tree parts: TRIAL (5%), TRAIN (45%) and TEST (50%). Following the RTE challenge [13], we keep the TEST portion unseen while using the rest of the data for development. The learning process, as described in [1], consists of three sub-learning components: improving the LLF generator, adding new facts to the KB and introducing new rules in the IR. The process is carried out manually while being facilitated with LangPro.

Analysis of false negatives (i.e. the non-neutral problems that are classified as neutral) reveals that often they are caused by inadequate CCG derivations. For instance, one of the most common mistakes the CCG parsers make is wrong PP-attachments. We design a couple of tableau rules to fix these mistakes. For example, the VP_PP converts a prepositional phrase from complements to modifiers and is used in problems like SICK-2171 (see Table 1). The closure rule PP_ATT, informally speaking, finds phrases like $((\mathbf{mix}\, A)\, \mathbf{in}\, B)$ and $(\mathbf{mix}\, (A\, \mathbf{in}\, B))$ equivalent, hence contributes to the proof of entailment pairs like SICK-4879.

[7] It was used as a benchmark for the RTE challenge [13] at SemEval-14: `http://alt.qcri.org/semeval2014/task1/`

ID	Gold/LangPro	Problem (T: text & H: hypothesis)
247	C/C	T: The woman is not wearing glasses or a headdress H: A woman is wearing an Egyptian headdress
344	N/C	T: An Asian woman in a crowd is not carrying a black bag H: An Asian woman in a crowd is carrying a black bag
410	E/E	T: A group of scouts are hiking through the grass H: Some people are walking
1696	E/E	T: A little cat is drinking fresh milk H: The milk is being drunk by a cat
1745	N/N	T: A man is pushing the buttons of a microwave H: A man is being pushed toward the buttons of a microwave
2171	E/E	T: An egg is being (cracked$_{\mathtt{pp,vp}}$ (into a bowl)$_{\mathtt{pp}}$) by a woman H: A woman is cracking an (egg$_{\mathtt{pp,n}}$ (into a bowl)$_{\mathtt{pp}}$)
3535	N/N	T: Someone is boiling okra in a pot H: Someone is being boiled with okra in a pot
3537	C/N	T: Nobody is cooking okra in a pan H: Someone is cooking okra in a pan
4443	N/E	T: A man is singing to a girl H: A man is singing to a woman
4879	C/C	T: There is no man ((mixing vegetables) (in a pot)$_{\mathtt{vp,vp}}$) H: A man is mixing (vegetables (in a pot)$_{\mathtt{n,n}}$)
5264	N/E	T: A person is folding a sheet H: A person is folding a piece of paper
8501	N/C	T: The person is not going into the water H: The man is going into the water

Table 1. Problems from TRIAL and TRAIN with gold and LangPro's answers

$$\frac{V_{\mathtt{pp},\alpha}\,(p^{\mathtt{IN}}_{\mathtt{np,pp}}\,D) : [\vec{C}] : \mathbb{X}}{p^{\mathtt{IN}}_{\mathtt{np},\alpha,\alpha}\,D\,V_\alpha : [\vec{C}] : \mathbb{X}}\text{VP\_PP} \qquad \frac{\mathbf{crack}_{\mathtt{pp,vp}}\,(\mathbf{into}^{\mathtt{IN}}_{\mathtt{np,pp}}\,b_e) : [c_e] : \mathbb{T}}{\mathbf{into}^{\mathtt{IN}}_{\mathtt{np,vp,vp}}\,b_e\,\mathbf{crack}_{\mathtt{vp}} : [c_e] : \mathbb{T}}$$

$$\frac{\begin{array}{r}[p^{\mathtt{IN}}\,b] : V\ : [d, \vec{C}] : \mathbb{F} \\ V : [d, \vec{C}] : \mathbb{T} \\ p^{\mathtt{IN}}_{\mathtt{np,pp}} : [b, d] : \mathbb{T}\end{array}}{\times}\text{PP\_ATT} \qquad \frac{\begin{array}{r}[\mathbf{in}^{\mathtt{IN}}_{\mathtt{pp,vp,vp}}\,b] : \mathbf{mix}_{\mathtt{np,vp}} : [d, c] : \mathbb{F} \\ \mathbf{mix}_{\mathtt{np,vp}} : [d, c] : \mathbb{T} \\ \mathbf{in}^{\mathtt{IN}}_{\mathtt{np,pp}} : [b, d] : \mathbb{T}\end{array}}{\times}$$

The dataset contains many expressions like *body of*, *group of*, *slice of*, etc. In order to correctly reason over the sentences involving these expressions, a new rule GRP_OF is introduced. Roughly speaking the rule asserts for *group members* whatever holds for a *group*. With the help of the rule it is possible to relate *group of scouts* to *people* in SICK-410 and prove H from T.

$$\frac{\begin{array}{r}G_{\mathtt{pp,n}}\,(\mathbf{of}^{\mathtt{IN}}_{\mathtt{np,pp}}\,b_e) : [c_e] : \mathbb{T} \\ V : [c_e] : \mathbb{X}\end{array}}{V : [b_e] : \mathbb{X}}\text{GRP\_OF} \qquad \frac{\begin{array}{r}\mathbf{group}_{\mathtt{pp,n}}\,(\mathbf{of}^{\mathtt{IN}}_{\mathtt{np,pp}}\,b_e) : [c_e] : \mathbb{T} \\ \mathbf{walk}_{\mathtt{vp}} : [c_e] : \mathbb{T}\end{array}}{\mathbf{walk}_{\mathtt{vp}} : [b_e] : \mathbb{T}}$$

where $G \in \{\mathbf{body}, \mathbf{group}, ...\}$

Acc% on TRAIN	Not alig.	Aligned	Both
RAL=50	75.24	80.80	80.84
Disj & RAL=50	76.87	80.87	80.93
Disj & RAL=800	-	**81.33**	**81.44**

(a) Results of the experiments on alignment and the disjoint relations. Employed LLFs are obtained from the C&C derivations.

Gold \ LangPro	E	C	N
Entailment	**802**	0	612
Contradiction	1	**501**	218
Neutral	25	7	2761

(b) The confusion matrix of LangPro (with the best configuration) on TEST. LLFs are obtained from C&C and EasyCCG derivations.

Table 2. Results of the experiment on TRAIN and the evaluation on TEST

4.2 Experiments

Most of the sentences in SICK are generated by altering other sentences. As a result, a text and a hypothesis often have at least one multiword chunk in common (e.g., SICK-2865 in Figure 3). In general, aligning T and H and neglecting shared phrases are commonly used by RTE systems. To test the effect of alignment, we add an optional aligner to the LLF generator. The aligner finds a set of compound terms shared by all the LLFs of a problem and treats them as constant terms, i.e. with no internal structure. Since the PE is not able to expand these constant terms, which is believed to be worthless, this increases chances for finding a proof. During experiment the prover is limited with 50 rule applications and three options are tested on TRAIN. The first row of Table 2a shows the results of the experiment. The prover performs more than 5% better with aligned LLFs in contrast to the non-aligned ones. Note that alignment process can make some information inaccessible for the prover that might be crucial for closing a tableau. For this reason, we also test a combined method where if a proof is not found with aligned LLFs then non-aligned ones are tried by the prover. The combined system shows little improvement over the aligned one.[8]

The hypernymy relation is very important for identifying contradictions and closing tableau branches. With respect to the dataset, it seems that WordNet provides the prover with sufficient hypernymy information. In Figure 2, it was shown how disjoint concepts can contribute to reasoning. Unfortunately, it is not clear how to get the high quality disjoint relations from WordNet in order to check the contribution of the relation to the prover. For this reason, for each problem in TRAIN we collected pairs of nouns and annotated 500 most frequent ones (with around 30% of coverage) on disjointness. The annotations were asserted as facts in the KB. The results in Table 2a show that only the system with non-aligned LLFs got the highest improvement (1.6%). Little improvement on the aligned system was expected as it treats some chunks as constants and is less effected with lexical knowledge than the non-aligned version. In case of the effective rule application limit (800) [1] the improvement over the aligned version still stays minor. This suggests that the rule application strategy of the prover rarely needs the information about disjoint concepts.[9]

[8] The reason of the improvement is the problems with prepositional phrases like *into a bowl* in SICK-2171. Aligning such kind of phrases prevents the tableau rules, e.g. VP_PP, to further unfold the internal structure of the phrase.

[9] After introducing the disjoint relations, more than 20 sentences in TRAIN obtained inconsistent meaning. For example, the hypothesis in SICK-3537 is parsed by C&C in such a way that it entails *someone is an okra*. The prover finds this meaning inconsistent as *okra* and *person* are disjoint concepts. As a result an inconsistent sentence is an indicator for wrong parsing. To avoid misclassification due to this kind of sentences, first each sentence is separately checked by the prover on consistency and afterwards the whole problem is analyzed.

TEST / LangPro+Disj	4927 problems		
	Prec%	Rec%	Acc%
Baseline (majority)	-	-	56.69
C&C 800 aligned	97.42	56.51	80.56
EasyCCG 800 aligned	97.60	57.22	80.88
Combined 800 aligned	97.52	60.73	82.34
C&C 800 both	97.43	56.84	80.70
EasyCCG 800 both	**97.61**	57.45	80.98
Combined 800 both	97.53	**61.06**	**82.48**

(a) Evaluation of the versions of LangPro (with disjoint facts) on TEST

Systems on TEST	Prec%	Rec%	Acc%
Illinois-LH	81.56	**81.87**	**84.57**
ECNU	84.37	74.37	83.64
UNAL-NLP	81.99	76.80	83.05
LangPro (Comb.)	**97.53**	61.06	82.48
SemantiKLUE	85.40	69.63	82.32
Meaning Factory	93.63	60.64	81.59
LangPro [1]	97.95	58.11	81.35
Nutcracker [19]	-	-	78.40

(b) Comparing LangPro to the top RTE systems of SemEval-14

Table 3. Evaluation results of LangPro on the SICK TEST

5 Evaluation

For the evaluation we use TEST which was held out during the adaptation period. In addition to the LLFs generated from the C&C derivations, we also employ the LLFs obtained from another CCG parser, EasyCCG.[10] To evaluate the quality of LLFs with different source parsers, first the prover with the two best configurations is evaluated for each type of LLFs. The results of the evaluation are given in Table 3a. According to each measure, the prover reasons better with LLFs based on EasyCCG compared to those from C&C. This effect is explained by the architecture of EasyCCG. As the parser uses probabilistic approach only for supertagging (e.i. assigning the CCG categories to lexical items), it makes more probable that very similar T and H will be tagged and later parsed similarly.

The best performance is achieved by the combined prover that finds a proof if either the C&C-based or EasyCCG-based provers find it (in case of the different positive answers from the provers, the combined one classifies a problem as neutral). An extremely high precision (>97%) of the prover on unseen problems is explained by its sound rules. The analysis of false positives over TRAIN reveals that most of the *mistakes* represent dubious cases (e.g., SICK-344,8501,5264) or often caused by the multi-senses from WordNet (e.g., SICK-4443). The confusion matrix in Table 2b shows that it is almost never the case that entailment and contradiction are confused by the prover. Better performance over contradiction problems is explained by the fact that decomposing entries with true signs is more efficient and respectively a tableau starts with true entries while checking a problem on contradiction.

LangPro with these results makes in top 5 of the SemEval RTE challenge [13] while still being a pure logic-based system in contrast to the rest of the systems.[11] It is true that in comparison to statistical RTE systems, our system is more brittle as it is sensitive towards errors from the parsers and lack of tableau rules. But it can still prove relations that state-of-the-art systems are not able to account for. For example, SICK-247, 1696 were wrongly classified by almost all systems in the top 5 of SemEval-14[12], but LangPro is able to prove them correctly. A tableau proof for SICK-1696 is given in Figure 4. Also our system avoids to classify problems like SICK-1745, 3535 as positive in contrast to each system in the top 4.

[10] Unlike the C&C derivations, the EasyCCG derivations were not used in the development of the LLF generator [1]. In this way, we check whether the generator generalizes enough well for other CCG parsers.

[11] The brother system of Nutcracker, the Meaning Factory [4], obtains similar results as our system, but it employs machine learning techniques and similarity measures.

[12] Except Meaning Factory that correctly guesses only SICK-1696.

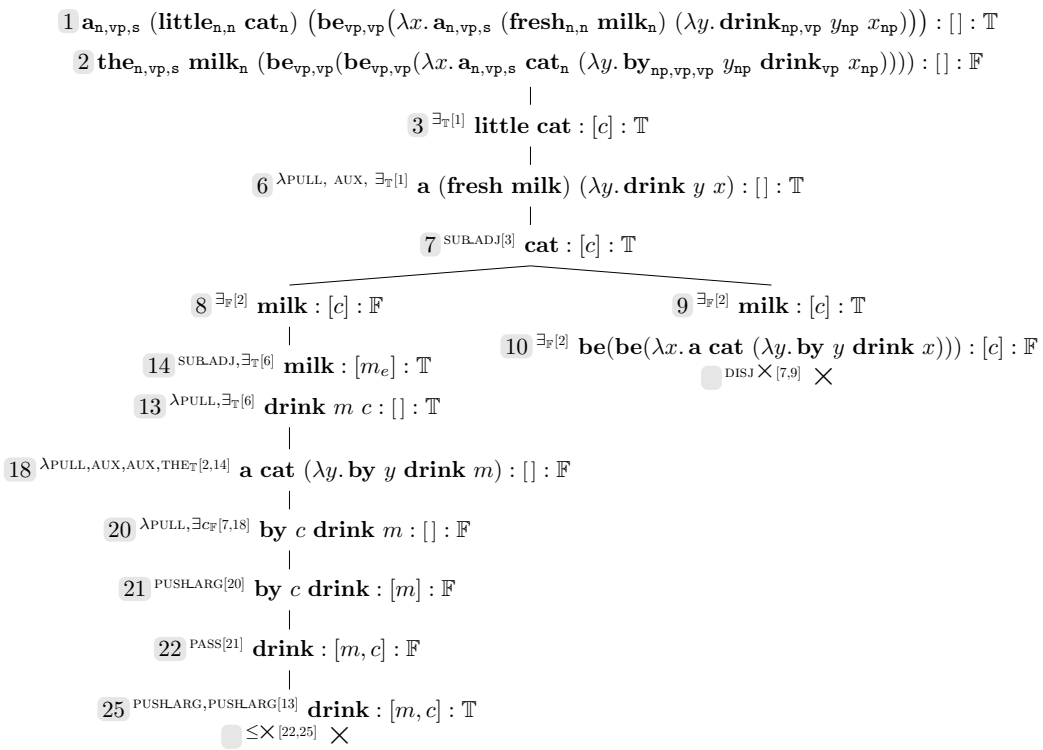

Figure 4. The tableau proving that SICK-1696 represents entailment. Some nodes from the initial tableau proof are omitted as they are not relevant for the proof.

6 Conclusion

We have presented the extended theory of the natural tableau and showed that it is viable theory for wide-coverage natural reasoning. The natural language theorem prover LangPro, based on that theory, achieves high competitive results on the SICK dataset while still being as reliable as theorem provers used to be for formal logics. The prover overcomes common shortcomings of most of the RTE systems nowadays on the market. It is fluent in reasoning over Boolean operators and quantifiers, has a quite expressive language where higher-order quantifiers or terms, like subsective adjectives, can be expresses and its reasoning skills are not limited to single-premised arguments. The theory and the prover are also able to provide a counterexample of an argument; every open tableau branch could offer a candidate for it.

A transparent decision procedure is another advantage of our model. A combination of LLFs, the forms similar to surface forms, and tableau rules, intuitively interpretable schematic rules, presents a suitable framework for studying human reasoning and information processing, e.g., to explain a complexity of a certain entailment in terms of a structure of a closed tableau.

In future, we plan to explore possibilities of improving the LLF generator since the performance of the system significantly hinges on the quality of LLFs. One of the possibilities is to check whether the 2nd or 3rd best derivations of EasyCCG contribute to better LLFs. Testing the prover on another RTE dataset, for example, those of the RTE challenges [7] or a more recent SNLI dataset [5], further challenges each component of the prover. For instance, taking into account characteristics of the annotation in the SNLI dataset, the prover needs a new device to anchor the events occurring in a text and a hypothesis.

References

[1] Abzianidze, L: A Tableau Prover for Natural Logic and Language. In the proceedings of EMNLP, ACL (2015)

[2] Abzianidze, L: Towards a Wide-coverage Tableau Method for Natural Logic. In Murata, T., Mineshima, K., Bekki, D. (eds.), New Frontiers in Artificial Intelligence, LNCS, vol. 9067, pp. 66–82. Springer Verlag (2015)

[3] Bos, J., Markert, K.: Recognising Textual Entailment with Robust Logical Inference. In: J. Quinonero-Candela, I. Dagan, B. Magnini, F. d'Alch-Buc (eds): Machine Learning Challenges, MLCW 2005, LNAI (3944), pp 404-426 (2006)

[4] Bjerva, J., Bos, J., Van der Goot, R., Nissim, M.: The Meaning Factory: Formal Semantics for Recognizing Textual Entailment and Determining Semantic Similarity. Proceedings of the 8th SemEval, pp 642-646 (2014)

[5] Bowman, S. B., Angeli, G., Potts, C., Manning, C. D.: A Large Annotated Corpus for Learning Natural Language Inference In the proceedings of EMNLP, ACL (2015)

[6] Clark, S., Curran, J.R.: Wide-Coverage Efficient Statistical Parsing with CCG and Log-linear Models. Computational Linguistics, 33(4) (2007)

[7] Dagan, I., Glickman, O., Magnini, B.: The PASCAL recognising textual entailment challenge. Machine Learning Challenges: Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment. LNCS, vol. 3944, pp. 177-190. Springer Berlin Heidelberg (2006)

[8] Fellbaum, Ch. (eds.): WordNet: an Electronic Lexical Database. MIT press (1998)

[9] de Groote, Ph.: Towards Abstract Categorial Grammars. In Proceedings of the 39th ACL Conference, pp. 148-155 (2001)

[10] Heim, I., Kratzer, A.: Semantics in Generative Grammar. Blackwell, Oxford (1998)

[11] Lewis, M., Steedman, M.: A* CCG Parsing with a Supertag-Factored Model. In Proceedings of the EMNLP 2014, pp. 990-1000. ACL (2014)

[12] MacCartney, B.: Natural Language Inference. Ph.D. Dissertation, Stanfor University, USA (2008)

[13] Marelli, M. *et al.*: SemEval-2014 Task 1: Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Textual Entailment. Proceedings of the 8th SemEval (2014).

[14] Marelli, M. *et al.*: A SICK Cure for the Evaluation of Compositional Distributional Semantic Models. In Proceedings of LREC, Reykjavik (2014)

[15] Montague, R.: English as a formal language. In: Linguaggi nella societa e nella Tecnica, pp. 189-224. Edizioni di Comunita (1970); reprinted in: Formal Philosophy; Selected papers of Richard Montague. Thomason, R. H. (eds.), pp. 108–221. Yale University Press (1974)

[16] Muskens, R.: Meaning and Partiality. CSLI, Stanford (1995)

[17] Muskens, R.: Language, Lambdas, and Logic. In: Kruijff, G., Oehrle, R. (eds.) Resource-Sensitivity, Binding and Anaphora. Studies in Linguistics and Philosophy, vol. 80, pp. 23–54. Springer, Heidelberg (2003)

[18] Muskens, R.: An Analytic Tableau System for Natural Logic. In: Aloni, M., Bastiaanse, H., de Jager, T., Schulz, K. (eds.) Logic, Language and Meaning. LNCS, vol. 6042, pp. 104–113. Springer, Heidelberg (2010)

[19] Pavlick, E., Bos, J., Nissim, M., Beller, C., Van Durme, B. & Callison-Burch, C.: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, pp. 1512–1522. ACL (2015)

[20] Winter, Y., Zwarts J.: Event Semantics and Abstract Categorial Grammar. In Kanazawa, M., Kornai, A., Kracht, M., Seki, H. (eds.) The Mathematics of Language. LNCS, vol. 6878, pp. 174-191. Springer, Heidelberg (2011)